REMARKS

Rejection of claims 1-19 under 35 U.S.C. §102(a) as being anticipated by Bohrer

The Examiner rejected claims 1-19 under 35 U.S.C. §102(a) as being anticipated by Bohrer. Each of these claims is addressed below.

Claim 1

Claim 1 recites:

> 1. An apparatus comprising:
> at least one processor;
> a memory coupled to the at least one processor;
> class configuration data comprising a plurality of entries residing in the memory, each class configuration entry including a key-value pair, wherein the key includes information relating to a selected processing context and the value includes configuration data for a class in the selected processing context; and
> an object oriented class replacement mechanism residing in the memory and executed by the at least one processor that generates an instance of a selected class by using a key that includes context information to access the appropriate entry in the class configuration data.

In rejecting claim 1, the Examiner states:

> Bohrer discloses . . . class configuration data comprising a plurality of entries residing in the memory, each class configuration entry including a key-value pair, wherein the key includes information relating to a selected processing context and the value includes configuration data for a class in the selected processing context (see fig. 5 and fig. 6, the key value pair here is the factory class and configuration data and class and the processing of the context of the class, col. 4, lines 50-59 and col. 9, lines 32-62);

Applicant acknowledges there are some similarities between the claimed invention and Bohrer. Bohrer represents the state of the art for class replacement as implemented in

IBM's San Francisco framework, which is discussed in detail in applicant's specification at p. 3 lines 5-26:

> To address this problem, a known method for class replacement has been implemented that uses class configuration data that is separate from and external to a class, and can thus be changed to create a new replacement class without rebuilding the program. This approach has been used in the San Francisco framework developed by IBM. The class replacement in the San Francisco framework uses keys known as tokens that are stored in a table of configuration data along with the class names that correspond to each token. When an instance of a class is needed, its token is passed to a factory object, which determines from the configuration data the class that corresponds to the token, and instantiates an instance of that class.
>
> One problem with the class replacement approach used in the San Francisco framework is that the configuration data is global to the object oriented system. Sometimes it is desirable to have class configurations that vary according to a specific processing context or environment. For example, a company may have different divisions, and each division may use its own specific way of performing currency conversions for transactions. These different divisions need a way to perform class replacement that is based to their processing environment, which is not currently supported in version 1.4.4 of the San Francisco framework. There are ways to retrieve context information in the San Francisco framework, but no way to base the class replacement according to this context information. Without an apparatus and methods for easily changing configuration data in an object oriented system to perform context-based class replacement, the computer industry will have to endure the limitations of prior art class replacement techniques.

Applicant's specification goes on to state at p. 9 line 23 to p. 10 line 6:

> As described in the Background section, the San Francisco framework provides a class replacement mechanism, but the configuration data for class replacement is global, and therefore cannot be based on a particular processing context. The San Francisco framework also allows retrieving context information, but this information cannot be used in the class replacement scheme. In addition, retrieving data such as policy objects associated with a context in the San Francisco framework typically requires communication with remote objects, which results in substantial performance penalties when retrieving the data.

Applicant's specification thus addresses class replacement as performed in Bohrer, and specifically states that the global configuration data in this type of a system does not allow class configurations to vary according to a specific processing context or environment. The key in differentiating the claimed invention from Bohrer lies in the fact that the claimed invention allows context-sensitive class replacement. Bohrer does not provide any class replacement that varies according to a processing context.

The Examiner's rejection of claim 1 is unclear. The Examiner states "the key value pair here is the factory class and configuration data and class and the processing of the context of the class." A key-value pair inherently includes two items, a key and a value. Yet the Examiner states that the key-value pair is 1) factory class; 2) configuration data and class; and 3) processing of the context of the class. Because the Examiner has mapped three elements of Bohrer on the key-value pair in the claims, it is unclear which elements map to the key and which map to the value. Because the Examiner has not clearly mapped a specific teaching in Bohrer to the key and has not mapped a specific teaching in Bohrer to the value of the key-value pair, applicant respectfully submits the Examiner has failed to establish a prima facie case of anticipation under 35 U.S.C. §102(a).

Bohrer does, in fact, teach key-value pairs. The keys correspond to the class tokens, while the values correspond to the class configuration data. A class token is used to locate its corresponding class configuration data. Thus, in FIG. 5 of Bohrer, the class token "employee" is passed to the Factory object 510. The Factory object then invokes the get_config() method on the Naming Service object (step 4), which returns the configuration date for the class corresponding to the "employee" token (step 5). This is described in detail at Bohrer col. 9 lines 14-32. Class replacement is accomplished in Bohrer by mapping the existing token "employee" to class configuration data for a new class that replaces the old class. This is described in detail in Bohrer at col. 9 lines 40-43.

Claim 1 expressly recites "wherein the key includes information relating to a selected processing context." Applicant respectfully asserts that the keys in Bohrer do not include "information relating to a selected processing context" as recited in claim 1. Claim 1 also recites that the class replacement mechanism generates an instance of a selected class "by using a key that includes context information to access the appropriate entry in the class configuration data." Applicant forcefully asserts that Bohrer has not teaching or suggestion of keys that include context information. To the contrary, the keys in Bohrer are class tokens, such as "employee". A simple class token does not include any information that relates to the processing context. For this reason, Bohrer does not teach keys that include information relating to a selected processing context, and Bohrer does not teach a class replacement mechanism that generates an instance of a selected class by using a key that includes context information to access the appropriate class configuration data. As a result, claim 1 is not anticipated by Bohrer. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 1 under 35 U.S.C. §102(a).

Claim 2

Claim 2 recites:

> 2. The apparatus of claim 1 wherein the key comprises context information appended to a class identifier.

In rejecting claim 2, the Examiner states: "Bohrer discloses where in the key comprises context information appended to a class identifier (container ID in this case: col. 8, lines 9-15)". The cited language of Bohrer states:

> Others may enforce access at the granularity of the whole persistent container, instances of specific classes, individual objects, or at the granularity that the persistent data store provides (such as at the table level for RDB). Again, the level of security is indicated in the container ID.

The cited language of Bohrer thus describes a container ID, but the container ID in Bohrer has nothing whatsoever to do with the class tokens that are used as keys. As a result, the container ID cannot read on the limitations in claim 2. Claim 2 states that the key comprises context information appended to a class identifier. The keys in Bohrer are class tokens, which are text identifiers of classes (such as "employee" shown in FIGS. 5 and 6). Nowhere does Bohrer teach or suggest keys that comprise anything appended to a class identifier. The token expressly taught in FIGS. 5 and 6 of Bohrer "employee" could be properly characterized as a class identifier, but there is no information appended to the class identifier that provides context information. For this reason, Bohrer does not read on the limitations in claim 2, and claim 2 is therefore not anticipated by Bohrer. Furthermore, claim 2 depends on claim 1, which is allowable for the reasons given above. As a result, claim 2 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 2 under 35 U.S.C. §102(a).

Claim 3

Claim 3 recites:

> 3. The apparatus of claim 2 wherein the class identifier comprises a class token that comprises a text string.

In rejecting claim 3, the Examiner states that Bohrer discloses these limitations, citing col. 7, lines 34-38, col. 9, lines 35-40, and fig. 4, item 302. Applicant readily admits that Bohrer discloses a class token that comprises a text string. Note, however, that claim 3 depends on claim 2. When read in light of the limitations in claim 2, the key in claim 3 comprises context information appended to a class token that comprises a text string. As stated above with respect to claim 2, nowhere does Bohrer teach appending anything to a class token. In fact, the class token "employee" shown in FIGS. 4 and 5 comprises only a class token, which expressly teaches away from the limitations in claims

2 and 3 that the key includes context information appended to a class identifier. Because the key in Bohrer does not include context information appended to the token, Bohrer does not anticipate claim 3. Furthermore, claim 3 depends on claim 2, which depends on claim 1, which is allowable for the reasons given above. As a result, claim 3 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 3 under 35 U.S.C. §102(a).

Claim 4

Claim 4 depends on claim 1, which is allowable for the reasons given above. As a result, claim 4 is allowable as depending on an allowable independent claim.

Claim 5

Claim 5 recites:

> 5. The apparatus of claim 1 further comprising a key generator mechanism that generates the key from a class identifier and from the context information.

In rejecting claim 5, the Examiner states that Bohrer discloses these limitations, citing fig. 5 for context of class information, citing the abstract, and col. 4, lines 1-10; col. 6 lines 57-67; and col. 7 lines 1-21. NONE of the cited portions of Bohrer teach ANYTHING regarding how the keys in Bohrer are generated. The keys in Bohrer are class tokens, such as "employee" shown in FIGS. 5 and 6, which are text strings that serve as labels. Because these are text strings, they are most likely assigned by a human programmer who decides what the name of the token for a particular class will be. Nowhere does Bohrer disclose a key that includes context information. Nowhere does Bohrer disclose how its keys are generated. As a result, it is impossible for Bohrer to disclose the key generator mechanism recited in claim 5. The key in claim 1 includes information relating to a

selected processing context. The key generator in claim 5 performs the processing that generates the key from a class identifier and from the context information. Because Bohrer does not disclose any key generator of any kind, it does not anticipate the key generator mechanism in claim 5. Furthermore, claim 5 depends on claim 1, which is allowable for the reasons given above. As a result, claim 5 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 5 under 35 U.S.C. §102(a).

Claim 6

Claim 6 recites:

> 6. A method for creating an instance of an object oriented class, the method comprising the steps of:
> (1) retrieving configuration data corresponding to the class in a selected processing context using a corresponding key that includes information relating to the selected processing context; and
> (2) instantiating the instance of the class using the retrieved configuration data.

In rejecting claim 6, the Examiner states that Bohrer discloses limitation (1) , citing fig. 5 and fig. 6, and col. 98 lines 15-32. The reference to col. 98 is apparently a typographical error, because there is no column 98 in Bohrer. While Bohrer does disclose retrieving configuration data corresponding to a key that is a class token, and instantiating the instance of the class using the retrieved configuration data, Bohrer has no teaching whatsoever regarding a key that includes information relating to the selected processing context, as recited in claim 6. For this reason, claim 6 is allowable over Bohrer. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 6 under 35 U.S.C. §102(a).

Claim 7

Claim 7 depends on claim 6, which is allowable for the reasons given above. As a result, claim 7 is allowable as depending on an allowable independent claim.

Claim 8

Claim 8 recites:

> 8. The method of claim 7 wherein the step of storing the configuration data with the corresponding key comprises the step of generating a key from a class identifier and from the context information.

In rejecting claim 8, the Examiner states that Bohrer teaches these limitations, citing col. 6 lines 57-67 and col. 7 lines 1-21. Nowhere in this cited language does Bohrer teach how its class tokens are generated. In addition, the keys in Bohrer do not include context information, so no step of generating a key from a class identifier and from context information can be reasonably inferred from Bohrer. Applicant respectfully asserts that Bohrer has no teaching whatsoever regarding the generation of its keys, and further assert that the keys in Bohrer are text strings that serve as labels and do not contain context information. For these reasons, claim 8 is allowable over Bohrer. Furthermore, claim 8 depends on claim 7, which depends on claim 6, which is allowable for the reasons given above. As a result, claim 8 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 8 under 35 U.S.C. §102(a).

Claim 9

Claim 9 recites that the key comprises context information appended to a class identifier, similar to the limitation in claim 2, which is discussed in detail above. Applicant respectfully asserts that claim 9 is allowable for the same reasons given above for claim 2. Furthermore, claim 9 depends on claim 6, which is allowable for the reasons given above. As a result, claim 9 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 9 under 35 U.S.C. §102(a).

Claim 10

Claim 10 contains limitations similar to claim 3, and is therefore allowable for the same reasons given above with respect to claim 3. Furthermore, claim 10 depends on claim 9, which depends on claim 6, which is allowable for the reasons given above. As a result, claim 10 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 10 under 35 U.S.C. §102(a).

Claim 11

Claim 11 contains limitations similar to those found in claims 5 and 8, and is therefore allowable for the same reasons given above with respect to these claims. Furthermore, claim 11 depends on claim 6, which is allowable for the reasons given above. As a result, claim 11 is also allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 11 under 35 U.S.C. §102(a).

Claim 12

In rejecting claim 12, the Examiner states that Bohrer discloses generating a key that includes information relating to the current processing context, citing the abstract and col. 4, lines 1-10. This cited language in Bohrer does not mention keys at all. As stated above in the discussion of claims 1, 5, 6, 8 and 11, Bohrer does not disclose how its keys are generated. From the simple text string "employee" shown in FIGS. 5 and 6, we can see that the keys in Bohrer do not contain any context information. Applicant strenuously asserts that the generation of a key that includes information relating to the current processing context is not taught or suggested in Bohrer. As a result, claim 12 is allowable over Bohrer. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 12 under 35 U.S.C. §102(a).

Claim 13

Claim 13 recites:

> 13. A program product comprising:
> an object oriented class replacement mechanism that generates an instance of a selected class by using a key that includes information relating to a selected processing context to access an appropriate entry in class configuration data stored external to the class; and
> signal bearing media bearing the object oriented class replacement mechanism.

The Examiner states that Bohrer teaches these limitations. However, Bohrer does not teach or suggest "a key that includes information relating to a selected processing context" as recited in claim 13. This limitation is discussed in detail above with respect to claim 1. Because Bohrer does not teach a key that includes information relating to a selected processing context, Bohrer does not anticipate claim 13. Applicant respectfully requests reconsideration of the Examiner's rejection of claim 13 under 35 U.S.C. §102(a).

Claims 14 and 15

Claims 14 and 15 depend on claim 13, which is allowable for the reasons given above. As a result, claims 14 and 15 are allowable as depending on an allowable independent claim. Applicant respectfully requests reconsideration of the Examiner's rejection of claims 14 and 15 under 35 U.S.C. §102(a).

Claims 16-19

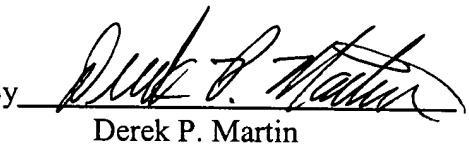
The Examiner rejected claims 16-19 based on the rationale for rejecting claims 2-5. Applicant asserts that claims 16-19 are allowable for the same reasons given above with respect to claims 2-5. Applicant respectfully requests reconsideration of the Examiner's rejection of claims 16-19 under 35 U.S.C. §102(a).

Conclusion

In summary, Bohrer does not teach, support, or suggest the unique combination of features in applicant's claims presently on file. Therefore, applicant respectfully asserts that all of applicant's claims are allowable. Such allowance at an early date is respectfully requested. The Examiner is invited to telephone the undersigned if this would in any way advance the prosecution of this case.

Respectfully submitted,

MARTIN & ASSOCIATES, L.L.C.
P.O. Box 548
Carthage, MO 64836-0548
(417) 358-4700

By 
Derek P. Martin
Reg. No. 36,595